IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:
Tormasov *et al.*
Appl. No.: 09/918,031
Filed: July 30, 2001
For: **Virtual Computing Environment**

Confirmation No.: 1032
Art Unit: 2145
Examiner: Mirza, Adnan M.
Atty. Docket: 2230.0400001/MBR/GSB

Declaration of Alexander Tormasov, Dennis Lunev, Serguei Beloussov, Stanislav Protassov and Yuri Pudgorodsky under 37 C.F.R. § 1.131

Commissioner for Patents
Washington, D.C. 20231

Sir:

The undersigned, Alexander Tormasov, Dennis Lunev, Serguei Beloussov, Stanislav Protassov and Yuri Pudgorodsky, declare and state that,

1. We are the inventors of the above-captioned application, U.S. Appl. No. 09/918,031, filed July 30, 2001.

2. Prior to August 24, 2000, we, the inventors, had completed our invention in a WTO country (specifically, in Russia), as claimed in the subject application, evidenced by the following:

3. Exhibit A, entitled "ASPcomplete Virtual Cluster Software Architecture," version 1.07 confirms the date of conception prior to the earliest priority date of Bandhole et al., U.S. Patent Publication No. 2002/0049803 (i.e., prior to August 24, 2000, the filing date of the provisional application to which Bandhole et al. claims priority). Specifically, page 2 of Exhibit A, which shows the revision history of the document, confirms the conception date at least as early as the April-May 2000 time frame.

4. Exhibit B, entitled "ve.log," represents a partial record of additional work that was directed to actual reduction to practice of the invention. Of particular relevance are the following items in the time frame between August 24, 2000 and February 16, 2001, with the dates highlighted in bold, and presented below in reverse chronological order for the Examiner's convenience:

```
revision 1.43.2.3
date: 2001/02/10 17:26:45;  author: den;  state: Exp;  lines: +20 -0
Compile time bugfixes
VE down support
----------------------------
revision 1.43.2.2
```

```
date: 2001/02/09 19:00:45;  author: den;  state: Exp;  lines: +34 -10
New 2-level routing scheme support
----------------------------
revision 1.43.2.1
date: 2001/02/02 22:08:36;  author: den;  state: Exp;  lines: +35 -8
veip hash support
----------------------------
revision 1.43
date: 2000/12/20 15:39:48;  author: den;  state: Exp;  lines: +16 -2
branches:  1.43.2;
for_each_task_ve performace tweak
is_ve_initialized call is removed (init tweak)
----------------------------
revision 1.42
date: 2000/12/19 16:13:40;  author: den;  state: Exp;  lines: +19 -11
NET update
----------------------------
revision 1.41
date: 2000/12/15 18:04:34;  author: yur;  state: Exp;  lines: +1 -1
typo fix
----------------------------
revision 1.40
date: 2000/12/13 12:55:42;  author: den;  state: Exp;  lines: +5 -2
Added VE IP into /proc interface
----------------------------
revision 1.39
date: 2000/12/05 15:33:11;  author: den;  state: Exp;  lines: +2 -3
CONFIG_VE_IRQ, CONFIG_VE_GLOBALSTATE defines removed
stack traces are removed when packet dropped
/proc/mount fix for VE
sys_syslog fix for VE
----------------------------
revision 1.38
date: 2000/11/15 14:52:54;  author: yur;  state: Exp;  lines: +3 -3
*** empty log message ***
----------------------------
revision 1.37
date: 2000/11/14 12:30:26;  author: yur;  state: Exp;  lines: +3 -3
Stupid memcpy fix
----------------------------
revision 1.36
date: 2000/10/25 14:29:03;  author: yur;  state: Exp;  lines: +3 -2
startup/uptime fix - once more
----------------------------
revision 1.35
```

```
date: 2000/10/17 14:45:30;  author: den;  state: Exp;  lines: +4 -2
SYSCTL fix
----------------------------
revision 1.34
date: 2000/10/16 12:42:33;  author: den;  state: Exp;  lines: +2 -1
CPU accouting added
boot race fix
----------------------------
revision 1.33
date: 2000/10/09 15:18:24;  author: den;  state: Exp;  lines: +4 -3
IP module fix
/proc/veinfo fix
----------------------------
revision 1.32
date: 2000/09/29 18:14:51;  author: den;  state: Exp;  lines: +10 -2
VE cleanup totally re-written
----------------------------
revision 1.31
date: 2000/09/25 15:10:41;  author: den;  state: Exp;  lines: +18 -4
devpts fix
----------------------------
revision 1.30
date: 2000/09/23 12:30:09;  author: yur;  state: Exp;  lines: +12 -1
2.4.0-test9pre6, reiserfs 3.6.17
----------------------------
revision 1.29
date: 2000/09/19 18:36:32;  author: den;  state: Exp;  lines: +12 -8
/proc patched
----------------------------
revision 1.28
date: 2000/09/18 09:16:16;  author: yur;  state: Exp;  lines: +2 -2
lock_kernel() for do_env_create()
----------------------------
revision 1.27
date: 2000/09/12 15:23:29;  author: den;  state: Exp;  lines: +1 -1
Packet dropper + VE0 process count fixup
----------------------------
revision 1.26
date: 2000/09/08 11:40:30;  author: den;  state: Exp;  lines: +25 -3
FairScheduler added
----------------------------
revision 1.25
date: 2000/09/01 15:10:59;  author: yur;  state: Exp;  lines: +2 -2
small fixes
----------------------------
```

```
revision 1.24
date: 2000/09/01 15:00:17;  author: den;  state: Exp;  lines: +12 -11
Compile fixes
----------------------------
revision 1.23
date: 2000/08/14 13:38:11;  author: den;  state: Exp;  lines: +28 -10
tty OOPS fix
revision 1.1.1.2
date: 2000/08/14 16:24:57;  author: yur;  state: dead;  lines: +0 -0
Removed errneous files from vendor branch
----------------------------
revision 1.1.1.1
date: 2000/08/14 14:00:27;  author: yur;  state: Exp;  lines: +480 -439
2.4.0-test6_reiserfs-3.6.12
```

5. Exhibits C, D and E are emails between one of the inventors and the patent attorney drafting the application, dated **1/08/2001**, **1/23/2001** and **1/27/2001**, relating to the preparation of the application (i.e., relating to constructive reduction to practice).

6. This application claims priority to U.S. Provisional Patent Application No. 60/209,255, filed February 16, 2001.

7. Thus, the invention was conceived prior to the earliest filing date of Bandhole et al., and the inventors were working diligently on constructive and/or actual reduction to practice between the filing date of Bandhole et al. and February 16, 2001, the priority date of this application.

8. As the persons signing below, we hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under § 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issue thereupon.

Dec 3 2004 Date	Alexander Tormasov
Dec 3 2004 Date	Dennis Lunev
Dec 3 2004 Date	Serguei Beloussov

Dec 3, 2004
Date

Stanislav Protassov

Dec 3, 2004
Date

Yuri Pudgorodsky

Exhibit A

ASPcomplete Virtual Cluster Software Architecture

Version 1.07

Revision History

Date	Ver	Description	Author
10 April 2000	1.0	First draft of the document created	Alexander Tormasov
20 April 2000	1.01	Write more items	Alexander Tormasov
21 April 2000	1.02	Start admin tools specifications	Alexander Tormasov
24 April 2000	1.03	Admin tools and possible configurations	Alexander Tormasov
27 April 2000	1.04	possible configurations corrected, remove VEstandalone info into separate file, new pic	Alexander Tormasov
30 April 2000	1.05	Rebuild all doc	Alexander Tormasov
3 May 2000	1.06	Correct terms, add comparisons for VE and FS (non-finished)	Alexander Tormasov
4 May 2000	1.07	comparisons DFS, usecases	Alexander Tormasov

Table of Contents

1. Introduction

This document describes ASPcomplete - Virtual Cluster design principles and some environments.

The goal of this project is to develop, maintain and deploy software solutions and policies that support scalable highly available platforms for Applications Service Providing. This support is transparent to the user and does not require modification of existing applications.

They key features that stand out ASPcomplete from similar solutions are:

- software-only cheap clustering solution based on free OS ASPLinux
- transparent support of virtually any Linux applications not working with dedicated hardware
- Virtual Environment approach to security and separation of end users of ASP platform with effective controllable sharing of resources
- dedicated distributed fault tolerance file system ASPFS
- cluster scalability from a couple of hardware nodes till hundred ones
- guaranteed level of Quality of Service for ASP platform users
- easies automated installation and maintenance procedures for hardware
- accounting and billing for ASP providers
- minimization of TCO both in software and hardware.

Our software is making available modern data and applications clustering technologies that allow a use of commodity computers for solving challenging computing problems for non-Linux industry specialists.

The purpose of this document is to provide comprehensive overview of the content of ASPcomplete Virtual Cluster (VC - Virtual Cluster) environment as a base for providing services for Application Service Providers (ASP) platform.

Below we use the following terms:

VE – Virtual Environment

VEFS – file system for VE (local version on the only computer)

VEutils – utilities to support VE operations

ASPFS – distributed transactional version of VEFS with replication

ASPstandalone – assembly of VE+VEFS+VEutils intended to run on the only computer (without fault tolerance and distributed replication features)

Virtual Cluster – a set of hardware nodes with VE+ASPFS+VEutils to provide a support for fault tolerated cluster

ASPcomplete – complete ASP platform solution including Virtual Cluster, support DB, Internet Gateway Layer, etc.

1.1 Recommended Reading

1.1.1 Books

- An introduction to distributed parallel computing, by Joel M. Crichlow (ISBN0131909681)
- UNIX Internals: The New Frontiers, by Uresh Vahalia (ISBN: 0131019082)
- In Search of Clusters, by Gregory F. Pfister *(ISBN: 0138997098)*
- High Performance Cluster Computing: Architectures and Systems, Vol. 1 and 2, by Rajkumar Buyya *(ISBN: 0130137847, 0130137855)*
- VMS File System Internals, by Kirby McCoy (ISBN 1555580564)
- VAXCluster principles, by Roy G. Daivs (ISBN 1555581129)
- The Mosix Distributed Operating System: Load Balancing for Unix, by Amnon Barak, et al. *(ISBN: 0387566635)*
-

1.1.2 Articles

-
- Selected Amoeba Papers, by A. Tanenbaum, et al. (http://www.cs.vu.nl/vakgroepen/cs/amoeba_papers.html)
- Protocols for file systems for clusters (http://www.inter-mezzo.org/docs/dfsprotocols.pdf)

1.1.3 Links

1.1.3.1 Must read

- File Systems for Clusters from a Protocol Perspective (http://www.extremelinux.org/activities/usenix99/docs/braam/braam.h
- Linux Virtual Server Project (http://linuxvirtualserver.org)
- Linux-HA Project Web Site (http://linux-ha.org/)
- LCC documents (ftp://people.redhat.com/wanger/clustering/docs/)
- Linux Enterprise Computing (http://linas.org/linux/)
- IP QoS Efforts (http://qos.ittc.ukans.edu/slides/)
- An API for Linux QoS Support (http://www.tisl.ukans.edu/~pramodh/courses/linux_qos/mainpage.htr
- Linux Soft Real Time project (http://www.uk.research.att.com/~dmi/linux-srt/)
- QLinux: A QoS enhanced Linux Kernel for Multimedia Computing (http://www.cs.umass.edu/~lass/software/qlinux/)
- ALTQ: Alternate Queueing (http://www.csl.sony.co.jp/person/kjc/programs.html)

1.1.3.2 Useful

- Linux High Availability HOWTO (http://metalab.unc.edu/pub/Linux/ALPHA/linux-ha/High-Availability-HOWTO.html)
- Creating Redundant Linux Servers (http://www.au.vergenet.net/linux/redundant_linux_paper/)
- HACC An Architecture for Cluster-Based Web Servers

(http://www.eecs.harvard.edu/~cxzhang/projects/hacc/final.html)

- Cluster Computing Homepage (http://www.tu-chemnitz.de/informatik/RA/cchp/index.html)
- MOSIX (http://www.mosix.cs.huji.ac.il/)
- Generic NQS (http://www.gnqs.org/)
- IEEE Parascope (http://computer.org/parascope/)
- Virtual Interface Architecture Draft Spec (http://www.viarch.org/)
- Short bibliography on QoS and traffic control in IP networks (http://www.info.fundp.ac.be/~obo/QoS/biblio/)
- References on CBQ (Class-Based Queueing) (http://www.aciri.org/floyd/cbq.html)
- Packet Scheduling Links (http://www.csie.nctu.edu.tw/~freedom/packet.html)

1.1.3.3 Other

- Rainfinity "A Technical Discussion of Rainwall Enterprise" (http://www.rainfinity.com/Products/whitepaper.pdf)
- Alinka - Raisin and Oranges (http://www.alinka.com/products.htm)
- ArrowPoint – Web Switching White papers (http://www.arrowpoint.com/solutions/white_papers/index.html)
- Java Apache Project "Fault tolerance with Apache JServ 1.1" (http://java.apache.org/jserv/howto.load-balancing.html)
- F5 Networks White papers (http://www.bigip.com/solutions/whitepapers/index.html)
- Eddie Project (http://www.eddieware.org/)
- The WALRUS Project (http://www.cnds.jhu.edu/walrus/)
- Resonate "Resonate Commander – Service Level Control" (http://www.intel-sol.com/products/resonate/commander.html)
- *To be continued...*

1.2 Overview

Here we want to give an overview of basic principles selected for a Virtual Cluster as a base for ASPcomplete solution.

To achieve a goals mentioned above ASPcomplete smoothly integrate Virtual Environments and dedicated distributed file system ASPFS into Virtual Cluster. Each end user of ASPcomplete receive his own virtual Linux with own root file system and could install any requested application – such as web server, office staff, network control packages, data base, etc. This approach guarantee not only a binary compatibility with modern Linux software but an appropriate level of Quality of Service (QoS) which could be provided by ASP platform.

1.2.1 Why virtual environment

Virtual environments (VE's) is a set of "OS inside OS" – full-featured Linux box being multiplied inside the only hardware unit. Each box could run inside virtually any Linux application (except ones working with specific hardware), have separate file system root files and effectively share resources of hardware (memory, CPU, disk, etc).

Effectiveness in CPU and memory (RAM) sharing achieved because of all processes inside are standard Linux processes handled by Linux kernel. Modern investigations and experiments in evaluation of amount of processes effectively running in Linux shows that it could be up to the at least 10K.

Another extremely important feature of VE is a real effectiveness of disk data sharing. All files share between different VE roots are the same from the file cache point of view and appears only once both in disk and RAM (as a shared code between processes running the same executable).

Security of VE is another advantage of VE. Users from different VE's are completely isolated from each other and cannot disturb each other.

1.2.2 Why distributed file system

Aim of *distributed file system* usually to provide an access to files from different instances of OS. Usually it guarantee some level of synchronization between different processes accessing the same data. In case of ASPcomplete solution distributed file system (called ASPFS) used for solution of slightly different tasks.

ASPcomplete should provide a service for end users – and each of them work inside his own VE with own set of files. Simultaneous access to files usually requested only in case of online backup of data. That means that we don't need complex and resource consuming full featured distributed lock manager.

But we need to be able to access this user data from any hardware node involved into cluster – just because any VE should be portable to start on any computer. In such a manner we could do load balancing of nodes and do maintenance operations requiring shutdown of them.

Fault tolerance is another significant feature of ASPcomplete solution. To provide it ASPFS should provide some level of logging and transactions – but without semantic contradictions to local FS (just because all applications running inside VE assume that this is a normal local file system).

There are different levels of fault tolerance supported by ASPLinuxCompete. The difference between them are the time of data actuality. Typically, normal level of them are about 30 sec – this means that in case of failure (of any kind – power failure of node or even applications failure) all data updated by end user will be consistent and with all modifications done later than 30 seconds before failure.

1.2.3 Why clustering

Term *clustering* has a very wide meaning in computing. Main idea of any cluster of computers is an ability to smoothly integrate computational (and not only) power of computers and their data-exchange facility into virtual centralized computational center. Key features of such a solutions are usually scalability and fault tolerance.

Most of the modern solutions in clustering usually require significant hardware support (utilization of expensive dedicated hardware).

ASPLinuxCompete integrate standard non-expansive hardware into solid highly scalable layered cluster to achieve declared goals of appropriate level of fault tolerance with minimal expenses.

Different configuration options make it suitable for wide range of ASP – from entry level up to cluster with hundreds nodes.

Important feature of ASPcomplete solution is an ability to smoothly integrate both software-only and hardware solutions, including dedicated hardware for RAID support, high performance multi CPU hardware nodes, rack – mount computers, etc.

1.2.4 Service Level Agreements and accounting

Service Level Agreements (SLA) is an important part of solution for ASP. It defines a way in which end user expect service from ASP. A SLA is a written agreement between the ASP and the business units (their clients), defining the nature and levels of service provided. Typically it should include a description of set of resources provided by ASP. Description include some static restrictions (like disk quota, fault tolerance level for file system ,etc) and some dynamic one (like network bandwidth). Also it includes policy of charging users for computer utilization – defining some kind of flat rate (with description of included services), penalties for over-usage of resources and other staff.

Problem of accounting and billing (charging of end user) is tightly coupled with SLA. To proper charge of end user ASP should correctly calculate resources utilization of hardware (of course in case when service cost depends upon them). That means that we have to install and maintain an accounting data base in which we store a history of end user operations. The requested performance usually defines by requested logging level. For instance in case of complete network traffic calculations information about any passing package could be stored.

For example, typical SLA for ISP hosting user web site is a flat rate for basic service and disk space with add-on for over usage of disk quota.

1.2.5 Quality of Service

Term Quality of Service (QoS) includes a wide range of utilization policy of computer hardware resources provided by underplaying OS. Depending of the SLA ASP platform should guarantee some minimal and maximal utilization of hardware, for example, minimal percentage of CPU used for particular VE or minimal network bandwidth.

Underlying OS supports not all possible restrictions. For example, CPU or virtual memory limitations are not supported in Linux right now.

In such a cases we have to avoid inclusion of non-supported features into SLA.

2. Virtual Cluster outline

2.1 ASPLinux Virtual Cluster environment

In general Virtual Cluster consist of (see picture 1):

1. ASP end users
2. Internet connection to ASP platform
3. Internet gateway layer -- hardware nodes running ASPLinux with firewall and distributed administration and billing Data Base
4. Internal network to connect to Virtual Cluster nodes
5. Virtual Cluster hardware nodes running ASPLinux with VE's
6. Internal high-speed network connected to every hardware node of dC to perform data redundancy and other operations with cluster data

2.1.1 ASP end users

There are a users using service from ASP. They have to order a service via web-interface from ASP, pay for it and receive the following information:

- Dedicated IP address (and DNS name) to connect to own VE
- UserID and password to login
- Screen delivery tools (like X server)
- Optional secure tools to connect to VE like Secure Shell – ssh client

2.1.2 Internet connection to ASP platform

Via this connection end-user have to connect securely to place where dC computers are placed. We assume that end user already have an Internet account from any ISP and able to connect to our Virtual Cluster.

Connection requirements:

- Reliable support of X11 protocol
- E-commerce grade security support (secure handshaking, channel encryption, data compression, etc)
- Easy-to use client installation to end user computer

2.1.3 Internet gateway layer (IGL)

Internet Gateway Layer consist of set of hardware nodes providing the following functionality:

- Traditional firewall support
- Network bandwidth monitoring/management
- Administration handling of environment
- Accounting Data base support to gather all information about running VE's and users

Logically all hardware unit seems as the only computer running as a gateway/firewall. For entry-level installation of Virtual Cluster it could be just the only computer.

Overall reliability of Virtual Cluster itself could not become better than the reliability of this layer.

2.1.4 Internal user-level network

Internal network should provide access from Virtual Cluster nodes to end users via firewall. Performance should be estimated taking into account the overall bandwidth of external Internet connection and internal service traffic.

2.1.5 Hardware nodes

This is a set of computers to provide main cluster functionality. They run an ASPLinux in VE edition with appropriate set of daemons supporting data clustering. Each node run a set of VE and has an access both to internal user-level network of cluster and high-speed data network. Amount of nodes depends upon a requested performance from ASP and could be vary from 2-3 till hundreds.

2.1.6 Internal high-speed data network

This network should serve all internal cluster data-exchange operations between nodes. In general it could be divided into parts in which all cluster operations should be performed.

The performance of this network depends upon an amount of hardware nodes and could be estimated before installation.

In some cases of low-end installation this network could be mixed with internal user-data network, but in general for complex clusters better to avoid such a mix.

2.2 Typical Virtual Cluster operations

Here we describe typical operations of Virtual Cluster. Remember, that cluster always stays in the only state (there are no "normal" and "fault-recovery" state as in other type of cluster).

From functional point of view all firewall/gateway part (IGL) logically could be treated as single computer.

2.2.1 ASPcomplete installation

2.2.2 New hardware node installation

2.2.3 Monitoring of resources

2.2.4 New virtual environment installation

Sequence of operations to create and install new VE in Virtual Cluster:

- User asks admin of database to create VE using self-administration tool (accessible from outside via web interface).

  User should provide the following information:
  - Max disk quota for user data (mandatory)
  - Pre-installed packages (mandatory)
  - Min performance of CPU unit (optional, mostly not used)

  - Min and max network bandwidth (optional)
  - Max memory usage (optional, not used in current version of kernel)
  - Availability level (mandatory – whether VE should be alive when no users coming in)
  - Max fault tolerance time for file system (mandatory)
  - Max living time for VE (mandatory – after this time all data will be erased)
  - Backup level (mandatory – how often all user data will be incrementally backupped)
  - Password data to access VE and access style (mandatory - telnet, ssh, X, etc, etc, etc)
  - Firewall options (mandatory – style of firewall to access VE via Internet)
  - Security monitoring level (optional – intruder monitoring, suspicious operations, etc, etc, etc)
  - Amount of IP addresses requested and their DNS names (optional – register in top level domains as .com)
  - Some payment information – VISA/etc/etc to pay for service and conditions of payment
- User receive as a respond from db the following information:
  - General information about requested and received service
  - IP information parameters (list of addresses/netmasks and DNS names, DNS server address, default router)
  - Optional client tools to login
  - Root password for logging in
  - Inside VE user receive an access to set of tools for administration of VE from inside and to access to billing DB to control his account

Admin also create a new mapping of VE environment (mostly references to appropriate VE template to avoid data doubling) and install some requested packages inside.

2.2.5 Environment runtime information gathering and control

ASP platform constantly gather information about running VEs and their processes. This should be done inside hardware node of Virtual Cluster. Control and gathering information about network utilization usually should be done in IGL (on firewall layer).

We could gather the following information:

- Elapsed CPU time for all processes (summarized) in VE – both user and system time

- Total size of virtual memory for all processes (summarized) in VE – both shared and not
- Amount of network packets (traffic) from all IP's belonging to each VE
- User-defined events (if any).

All these information are placed into accounting DB. Information gathering usually done on regular basis with rate about once per second.

Content of the accounting record in DB:

1. VE id
2. current time
3. elapsed CPU user time for all processes since last acquire
4. elapsed CPU system time for all processes since last acquire
5. current usage of virtual memory for all processes in VE
6. user data information (variable length)
7. network traffic since last acquire for all addresses (variable length field).

All network information stored here because it should be calculated inside VE, not on IGL (firewall). Otherwise we loose all traffic running between VE's.

2.2.6 Environment start/restart

To start environment special network daemon running in each Virtual Cluster hardware node (VEadmin) should receive a request from main DB. This request should have the following parameters:

- VEid
- Start mode (start-recover-restart)
- Fault tolerance level
- Restrictions set (min CPU%, virtual memory, network bandwidth, etc)

Then VEadmin daemon mount appropriate version of file system in /VE catalog from Virtual Cluster. After that it starts a script to create init process with new VEid, and run all scripts from normal startup places (like /etc/rc.d/...) from appropriate place in mounted file system.

2.2.7 Environment maintenance

Initial installation procedure typically the same for all VE except some list of additional packages requested while environment creation.

Mostly here we try to describe backup/restore procedures in Virtual Cluster. Because of the nature of our file system we could do online backup in any moment of time – just stopping garbage collection procedure and than fix backup time. Actuality of all data is guaranteed by our transaction algorithm of data modifications (we even don't need any locking to work with files simultaneously with VE).

Backup operations could be done in some dedicated VE simultaneously running with VE to be backup.

Restore procedure usually should not performed parallel with actual VE.

2.2.8 Fault recovery

In case of Virtual Cluster node failure we always has an actual version of all VE data. We have to start all VE's from failed node again on another node, mount VE file system and restart an environment. All recovering from failure in applications level should be done by application itself – we just guarantee some level of data actuality and consistency.

In case when end user makes some fatal mistake preventing him from proper access to VE we could run VE in fail-safe mode. In this mode we just start pre-defined template of "recovery VE" and mount actual failed VE file system. Then user login into this VE, fix his data and then restart his own just fixed VE.

2.3 Administration tools

They are intended to operate Virtual Cluster and all environments. There are the following classes of Administration tools:

- End user tools to work from inside VE
- VE user self- maintenance tools via web interface (creation, modification, bills paid/accounting state, etc)
- VE administration tool from IGL, including network administration
- Virtual Cluster general maintenance tools (including backup, nodes update, security monitoring, etc)

2.3.1 End user control tools

These tools are intended mostly to ASP platform end users to interact with ASP platform to receive some information about his VE state from inside VE (usually to use in shell scripts).

Typical task to solve using these tools:

I. Obtaining configuration information from Virtual Cluster db:
   - contract information (user information)
   - IP parameters (address, netmask, gateway, DNS server, DNS name)
   - disk quota (limit)
   - CPU % quota (limit)
   - virtual memory quota (limit)
   - network bandwidth (limit in kb/s or in/out traffic limit)

II. Obtaining current state information from Virtual Cluster db:
- disk quota usage
- network usage
- CPU usage
- virtual memory usage

All information should be received from VEinfo utility. Command line interface could be invoked interactively or in form

VEinfo –c "command".

2.3.2 End user self- maintenance tools

These tools are intended mostly to ASP platform end users to interact with ASP platform to create environment, change type of some services received from ASP administration, pay for it, check his account, etc. They have web-based interface – user connect to main web server of ASP platform via secure connection (like browsers with SSL support). Specifications for web screens see in Appendix.

2.3.3 VE handling tools

All these tools should run outside of user VE (in 0 VE usually on Virtual Cluster node). These tools are intended mostly to ASP platform administrator.

There are 2 kinds of such a tools in Virtual Cluster:

I. Tools to handle running environment on the same Virtual Cluster hardware node.

II. Tools to manipulate end-user data in accounting database.

Requirements to I kind of tools:

- Start/stop VE using information from admin data base
- Change running VE restrictions (CPU %, virtual memory, disk quota)
- Monitor VE (collect billing information, security monitoring, etc)

Requirements to II kinds of tools:

- Operate accounting DB:
    - Create/delete environment
    - Modify restrictions
    - Modify QoS parameters, including network bandwidth and firewall parameters
    - Check billing information to charge end user
- Reflect changes in DB to running VE and Internet Gateway Layer

2.3.4 Virtual Cluster maintenance tools

3. Virtual Cluster use cases

There are a couple of possible configurations available for a Virtual Cluster.

Particular configuration depends upon a requirements to applications running inside, amount of hardware nodes in cluster, reliability, handled data volume, etc.

3.1 Minimal (1 hardware unit configuration) – ASPstandalone

Strictly speaking, this is not a cluster at all, just one-computer installation of some tools from full cluster.

More information about please checks in ASPstandalone description document.

3.2 Web hosting ISP platform

Here we describe a typical configuration for mid-range Internet Service Provider (ISP) used mostly for web hosting (E-commerce, Internet shops, etc).

Theoretically, in case of web-only hosting seems that apache itself could do required multiplication. But from the security point of view VE approach is much more attractable for end users (just because VE completely isolate it from each other).

We estimate an amount of webs to be hosted by about 1000. Typically, most of them (800) are lightweight apache-based solutions that could be hosted on the same hardware node.

Rest of web servers requires more intensive operations (complex CGI processing, db gateway, etc) and should be placed on 2-3 hardware nodes.

Estimation of data update rate shows that for normal web most changeable part is logs (access, error, etc). In typical rate of 1000 hits per day (low rate) we have about 800000 hits per day (10 per sec). Every log record is about 100b – so we have total growth 1K bytes/sec. This should be small in most cases (even if we take into account other logs).

For intensive web apps it could be about 10 times worse – but still acceptable (max bandwidth for redundant data update for P2 300 is about 5Mb/sec).

So, typical configuration is:

IGL – 2 functionally mirrored non-expensive computers with firewall connected to Internet and to hardware nodes by 100BASE T network

4 hardware nodes with VE's (800 on one of them, 200 on others) – typical servers

1 highly stable hardware node with support DB and backup unit.

All connected into 2 segments of 100mb local network.

3.3 CPU - intensive applications hosting

Here we describe a typical configuration for provider of CPU-intensive applications with 1-minute availability time (service could be interrupted for not more than 1 min in case of any problems).

To be written.

4. Virtual Cluster comparisons

Here we make an overview of comparable solutions in ASP providing.

4.1 Virtual environment – like implementations

Here we describe typical server platforms analogous to proposed VE-based solutions.

	ASPcomplete	*ServerXchange*	*VMWare*	*BSD jail*
Project Goal	Complete ASP platform	Complete ASP platform	Virtual computer suitable to run foreign OS	Sandbox for dangerous applications
Implementation	Uni-kernel	Multiple kernels	Multiple kernels	Uni-kernel
Isolation level	Linux capabilities, namespace separation	Unknown	Virtual hardware	Access control filters
Performance loss	Very low, gains in some areas possible	Unknown	Low for CPU bound, high for IO bound	Very low
Resource Sharing	All	Unknown	No	All
Scalability	High	Medium	Low	High
QoS support	By OS, some additions planned	Yes	No	By OS
Fault tolerance	Fault isolation, automatic restart, transactions replication	Fault isolation	Fault isolation	Fault isolation
Administration Tools	Planned	Yes	Limited	No public, present third party providers
Billing Support	Planned	No	No	No

4.2 Distributed File System implementations

Here we compare Linux implementation of Distributed File Systems.

	Sharing	High Avail	Caching	Naming	Protection
NFS	Potentially inconcistent cache	Read-only replication	Negligible Client-side caches	Client Mount Point	UNIX identity, UNIX mode bits
AFS	Distributed locks Open/Close sessions	Read-Only replication	Large client caches	UNIFORM: Server controlled remote mount	Kerberos identity, ACLs on folders
Coda	Distributed locks Write Conflicts resolving inconsistency repair	Full Server replication, Disconnect Client operations	Large client caches	UNIFORM: Server controlled remote mount	Kerberos identity, full ACLs
AIX GPFS	Distributed locks	Local and Remote full replication	Large coherent local cache	UNIFORM in cluster	?
DCE/DFS	Distributed locks	Read-only replication			Encrypted identity, full ACLs, tokens
Intermezzo	currently none	currently none	volume replication write-back cache	UNIFORM	currently none
VAXClusters	Distributed locks	?	?	UNIFORM: resource manager	?
TorFS	yes	Full Server replication, persistent file versions	Dedicated data migration cache model for network traffic optimization	UNIFORM across cluster	access tokens (signatures), UNIX mode bits
VERFS	no	Full data transaction replication,	local FS cache, copy-on-	Separate for each virtual	UNIX identity,

		persistent file versions	write sharing	environment	UNIX mode bits

Continuation of comparison table

	Performance	Scalability	Typical applications	Uniq Features
NFS	Fair	10+ nodes	Client-Server file sharing	
AFS	Good	1000+ nodes	Uniform distributed namespace	
Coda	Good	1000+ nodes	uniform distributed namespace	Disconnected Client Operations
AIX GPFS	Very Good	100+ nodes	shared FS for cluster	data stripping, local and remote redundancy
DCE/DFS	?	10+ nodes	enterprise file sharing	
Intermezzo	?	?	research project	Disconnected Ops, write-back cache
Lustre	Good	10+ nodes	trusteed cluster filesystem	Complex recovery after node failure
TorFS	TBE	TBE	uniform distributed namespace across public networks	replication algorithm, strong encryption for each transaction
VERFS	TBE	TBE	High availability Virtual Environments	Local FS template sharing, copy-on-write cache, data transation replication

5. Quality of Service

To be written

6. Service Level Agreements and accounting

To be written

7. Appendix A

7.1 VEinfo specifications

This utility provide command-line interface to Virtual Cluster db and should be used by end user from inside running VE.

Configuration info for VEinfo should be placed in /etc/VE/VEinfo.conf (mostly parameters to call DB) and initialized before starting of VE.

Without parameters this executable returns VEid – VE identifier in form "NNNNNNNN" where N – hex digit.

Commands could be the following:

Command (stdin or param)	**Return** (stdout)
IP address ranges number	Decimal number N (N >= 1)
IP address N	Dotted decimal IP address (x.x.x.x) – interpreted as a start address in range or "ERROR" if not exist
IP netmask N	Dotted decimal IP mask (x.x.x.x) – interpreted as a netmask for address N or "ERROR" if not exist
IP default gateway	Dotted decimal IP address (x.x.x.x)
DNS server N	Dotted decimal IP address of DNS server number N (N >= 1) or "ERROR" if not exist
User information	Text string which user enter while VE creation
Disk quota	Decimal number in bytes
CPU quota	Decimal digit in Bogomips – interpreted as a minimal amount of CPU used exclusively by all VE processes
Virtual memory quota	Decimal digit in Mb – interpreted as a maximum of sum of virtual memory of all processes of VE
Network bandwidth quota	Decimal digits interpreted as a bytes-per-sec minimal bandwidth in both directions assigned to VE or 0 if not assigned
Network inbound bandwidth quota	Decimal digits interpreted as a bytes-per-sec minimal inbound bandwidth assigned to VE or 0 if not assigned
Network outbound bandwidth quota	Decimal digits interpreted as a bytes-per-sec minimal outbound bandwidth assigned to VE or 0 if not assigned
Network `traffic quota	Decimal digits interpreted as a network traffic limit in both directions for VE or 0 if not assigned

Network inbound traffic quota	Decimal digits interpreted as a network inbound traffic limit for VE or 0 if not assigned
Network outbound traffic quota	Decimal digits interpreted as a network outbound traffic limit for VE or 0 if not assigned
Current disk usage	Decimal number in bytes
Disk usage history starting from ddmmyyyyhhmm in N <minutes|hours|days>	Return appropriate amount of text string. In each string there are the only decimal digit in bytes of disk usage.
Current CPU usage	Decimal number in bytes interpreted as bogomips used by all processes of VE
CPU usage history starting from ddmmyyyyhhmm in N <minutes|hours|days>	Return appropriate amount of text string. In each string there are the only decimal digit in bytes of elapsed time incrementally.
Current virtual memory usage	Decimal number in Mbytes
Virtual memory usage history starting from ddmmyyyyhhmm in N <minutes|hours|days>	Return appropriate amount of text string. In each string there are the only decimal digit in Mbytes of vmem usage.
Inbound traffic	Decimal number in bytes – total traffic from measurement starting (usually VE creation moment)
Outbound traffic	Decimal number in bytes – total traffic from measurement starting (usually VE creation moment)
Overall traffic	Decimal number in bytes – total traffic from measurement starting (usually VE creation moment)
Inbound traffic history starting from ddmmyyyyhhmm in N <minutes|hours|days>	Return appropriate amount of text string. In each string there are the only decimal digit in bytes of appropriate traffic incrementally
Outbound traffic history starting from ddmmyyyyhhmm in N <minutes|hours|days>	Return appropriate amount of text string. In each string there are the only decimal digit in bytes of appropriate traffic incrementally
Overall traffic history starting from ddmmyyyyhhmm in N <minutes|hours|days>	Return appropriate amount of text string. In each string there are the only decimal digit in bytes of appropriate traffic incrementally

Before each command user could specify prefix "VEid NNNNNNNN" where NNNNNNNN is an identifier of VE.

7.2 VEutil specifications

VEutil intended to provide interface to VE-specific features of ASP Linux. Typically user run it in VE0 to create or control another environments. This is a command-line utility.

List of VEutil comman line options:

--create <VEid> <ip>

creates environment with the following id and IP address. Interface should be manually brought up inside VE by ifconfig (preferrably in rc.d scripts)

--verify

prints current environment id

--kill <VEid>

runs 'init 0' inside VE, kills the remnants in 30 sec and cleans
memory allocated for VE structures

--enter <VEid> [shell]

invokes unix shell (by default 'bash') with supplied <VEid>. This option is like a backdoor as check is made only for CAP_SETVEID capability

--perm <VEid> {--chr –blk} <major> <minor> <mask>

this call manipulates permissions for devices access checks BEFORE any filesystem checks. Rules applied in the following order:

<major> <minor>
<major> 0
0 0

7.3 End user web interface

Here we describe screens for browser-based control tools to provide communications between end user and ASP platform.

7.3.1 VE creation screen

It should consists of the following fields:

Basic screen:

- Max disk quota for user data (mandatory) – decimal number in Mb
- Availability level (mandatory – whether VE should be alive when no users coming in) – Boolean – true/false
- Max living time for VE (mandatory – after this time all data will be erased) – decimal number in days
- Password data to access VE and access style (mandatory - telnet, ssh,

X, etc, etc, etc) – 1 char string with root password and set of check boxes for: "telnet", "ssh", "rsh".

Advanced screen:

- Min performance of CPU unit (optional, mostly not used) – decimal number in bogomips
- Min and max network inbound bandwidth (optional) – 2 decimal numbers in kb/s – 0 if avoided
- Min and max network outbound bandwidth (optional) – 2 decimal numbers in kb/s – 0 if avoided
- Min and max network bandwidth (optional) – 2 decimal numbers in kb/s – 0 if avoided
- Min and max network quota (optional) – 2 decimal numbers in Mb's – 0 if avoided
- Min and max network inbound quota (optional) – 2 decimal numbers in Mb's – 0 if avoided
- Min and max network outbound quota (optional) – 2 decimal numbers in Mb's – 0 if avoided
- Max memory usage (optional, not used in current version of kernel) – decimal digit in Mb
- Max fault tolerance time for file system (mandatory) – decimal number in sec
- Backup level (mandatory – how often all user data will be incrementally backupped) decimal number in hours
- Firewall options (mandatory – style of firewall to access VE via Internet) – radio button with the following options:
  "open" - will allow anyone in
  "client" - will try to protect just this machine
  "simple" - will try to protect a whole network behind
  "closed" - totally disables IP services except secure loggin in
  "none" - disables the loading of firewall rules
  "filename" - will load the rules in the given filename (full path required)
- Security monitoring level (optional – intruder monitoring, suspicious operations, etc, etc, etc) – not implemented yet
- Amount of IP addresses requested and their DNS names in strings interpreted as DNS names for each IP address
  (optional – register in top level domains as .com – not implemented)
- Some payment information – VISA/etc/etc to pay for service and conditions of payment – not implemented

    - Pre-installed packages (mandatory) – not implemented
- User receive as a respond page with the following information:
    - General information about requested and received services
    - VEid, time of VE user data expiration
    - IP information parameters (list of addresses/netmasks and DNS names, DNS server address, default router)
    - Allowed client tools names to login (telnet, ssh, rsh).
    - Root password for logging in and later for access to his VE information

7.3.2 VE user self-maintenance screen

Here user receives information about parameters of his VE.

Request screen with the following strings:

- VEid in string
- Password

Respond screen (temporary all information in read-only mode because user could not change it):

- List of IP addresses with masks and DNS names
- Default gateway
- DNS server
- User information
- Disk usage/disk quota
- Total CPU usage/total CPU quota
- Virtual memory usage/quota
- Network bandwidth (in, out, sum)

Exhibit B

ve.log

```
RCS file: /home/1/cvs/virtuozzo/oldlinux/kernel/ve.c,v
Working file: virtuozzo/oldlinux/kernel/ve.c
head: 1.58
branch:
locks: strict
access list:
symbolic names:
        ubc_merge_den_001: 1.53
        era-001: 1.51
        test-2_4_1-0: 1.46
        main-ubc-14: 1.44
        sched-devel: 1.43.0.2
        ve_0_5_singtel: 1.43
        beforeMajorNetUpdate: 1.39
        beta_0_4_2: 1.23
        beta_0_4_1: 1.19
        beta_0_4: 1.17
        test4: 1.11
        alpha0_4: 1.9
        alpha_0_3: 1.8
        vendor: 1.1.1
        alpha_1: 1.2
        ve-before-yur: 1.2
keyword substitution: kv
total revisions: 66;    selected revisions: 66
description:
----------------------------
revision 1.58
date: 2001/04/11 10:51:40;  author: den;  state: Exp;  lines: +3 -14
major net cleanup + separated device chains
----------------------------
revision 1.57
date: 2001/03/28 10:14:17;  author: den;  state: Exp;  lines: +2 -2
compilation fix
----------------------------
revision 1.56
date: 2001/03/19 13:19:22;  author: den;  state: Exp;  lines: +1 -1
compile fix
----------------------------
revision 1.55
date: 2001/03/18 12:13:00;  author: den;  state: Exp;  lines: +5 -5
compilation fix for CONFIG_VE undefined
----------------------------
revision 1.54
date: 2001/03/18 11:23:06;  author: den;  state: Exp;  lines: +14 -7
eth0 visiblity fix
----------------------------
revision 1.53
date: 2001/03/09 18:50:50;  author: den;  state: Exp;  lines: +33 -3
ve_ip_map syscall implementation
----------------------------
revision 1.52
date: 2001/03/09 15:51:03;  author: den;  state: Exp;  lines: +1 -12
VENET double initialization fix
----------------------------
revision 1.51
date: 2001/03/07 16:09:01;  author: den;  state: Exp;  lines: +5 -5
pts hangup fix
----------------------------
revision 1.50
date: 2001/03/07 15:05:40;  author: den;  state: Exp;  lines: +5 -1
startup hang fix
```

```
----------------------------
revision 1.49
date: 2001/03/06 17:31:51;  author: den;  state: Exp;  lines: +1 -1
useless compilation fix: VECALLS undefined at all
----------------------------
revision 1.48
date: 2001/03/06 17:12:39;  author: den;  state: Exp;  lines: +27 -1005
*** empty log message ***
----------------------------
revision 1.47
date: 2001/03/03 16:13:12;  author: den;  state: Exp;  lines: +5 -1
venet device module support
----------------------------
revision 1.46
date: 2001/02/27 09:56:47;  author: den;  state: Exp;  lines: +6 -0
Network device separation
----------------------------
revision 1.45
date: 2001/02/21 14:18:14;  author: den;  state: Exp;  lines: +1 -0
process numbering in VE0 patch
----------------------------
revision 1.44
date: 2001/02/19 15:50:27;  author: den;  state: Exp;  lines: +85 -14
New scheduling code - merge from sched-devel
2-level routing - merge from sched-devel
----------------------------
revision 1.43
date: 2000/12/20 15:39:48;  author: den;  state: Exp;  lines: +16 -2
branches:  1.43.2;
for_each_task_ve performace tweak
is_ve_initialized call is removed (init tweak)
----------------------------
revision 1.42
date: 2000/12/19 16:13:40;  author: den;  state: Exp;  lines: +19 -11
NET update
----------------------------
revision 1.41
date: 2000/12/15 18:04:34;  author: yur;  state: Exp;  lines: +1 -1
typo fix
----------------------------
revision 1.40
date: 2000/12/13 12:55:42;  author: den;  state: Exp;  lines: +5 -2
Added VE IP into /proc interface
----------------------------
revision 1.39
date: 2000/12/05 15:33:11;  author: den;  state: Exp;  lines: +2 -3
CONFIG_VE_IRQ, CONFIG_VE_GLOBALSTATE defines removed
stack traces are removed when packet dropped
/proc/mount fix for VE
sys_syslog fix for VE
----------------------------
revision 1.38
date: 2000/11/15 14:52:54;  author: yur;  state: Exp;  lines: +3 -3
*** empty log message ***
----------------------------
revision 1.37
date: 2000/11/14 12:30:26;  author: yur;  state: Exp;  lines: +3 -3
Stupid memcpy fix
----------------------------
revision 1.36
date: 2000/10/25 14:29:03;  author: yur;  state: Exp;  lines: +3 -2
startup/uptime fix - once more
----------------------------
```

```
revision 1.35
date: 2000/10/17 14:45:30;  author: den;  state: Exp;  lines: +4 -2
SYSCTL fix
----------------------------
revision 1.34
date: 2000/10/16 12:42:33;  author: den;  state: Exp;  lines: +2 -1
CPU accouting added
boot race fix
----------------------------
revision 1.33
date: 2000/10/09 15:18:24;  author: den;  state: Exp;  lines: +4 -3
IP module fix
/proc/veinfo fix
----------------------------
revision 1.32
date: 2000/09/29 18:14:51;  author: den;  state: Exp;  lines: +10 -2
VE cleanup totally re-written
----------------------------
revision 1.31
date: 2000/09/25 15:10:41;  author: den;  state: Exp;  lines: +18 -4
devpts fix
----------------------------
revision 1.30
date: 2000/09/23 12:30:09;  author: yur;  state: Exp;  lines: +12 -1
2.4.0-test9pre6, reiserfs 3.6.17
----------------------------
revision 1.29
date: 2000/09/19 18:36:32;  author: den;  state: Exp;  lines: +12 -8
/proc patched
----------------------------
revision 1.28
date: 2000/09/18 09:16:16;  author: yur;  state: Exp;  lines: +2 -2
lock_kernel() for do_env_create()
----------------------------
revision 1.27
date: 2000/09/12 15:23:29;  author: den;  state: Exp;  lines: +1 -1
Packet dropper + VE0 process count fixup
----------------------------
revision 1.26
date: 2000/09/08 11:40:30;  author: den;  state: Exp;  lines: +25 -3
FairScheduler added
----------------------------
revision 1.25
date: 2000/09/01 15:10:59;  author: yur;  state: Exp;  lines: +2 -2
small fixes
----------------------------
revision 1.24
date: 2000/09/01 15:00:17;  author: den;  state: Exp;  lines: +12 -11
Compile fixes
----------------------------
revision 1.23
date: 2000/08/14 13:38:11;  author: den;  state: Exp;  lines: +28 -10
tty OOPS fix
----------------------------
revision 1.22
date: 2000/08/11 16:52:00;  author: den;  state: Exp;  lines: +42 -12
UNIX98 ptys support
----------------------------
revision 1.21
date: 2000/08/11 16:26:37;  author: den;  state: Exp;  lines: +34 -22
TTY work in progress
----------------------------
revision 1.20
```

```
date: 2000/08/11 15:48:57;  author: den;  state: Exp;  lines: +84 -71
TTY work in progress
----------------------------
revision 1.19
date: 2000/08/07 15:08:00;  author: yur;  state: Exp;  lines: +2 -2
2.4.0-test6-pre6 merged
----------------------------
revision 1.18
date: 2000/07/31 07:56:32;  author: den;  state: Exp;  lines: +0 -1
VE options are moved from Processor types/features to Kernel hacking
----------------------------
revision 1.17
date: 2000/07/25 16:01:10;  author: yur;  state: Exp;  lines: +1 -1
Copyright to SWSoft
----------------------------
revision 1.16
date: 2000/07/20 16:14:49;  author: den;  state: Exp;  lines: +3 -19
VE IPC cleanup procedures are added
----------------------------
revision 1.15
date: 2000/07/19 12:11:42;  author: den;  state: Exp;  lines: +5 -0
Patch cleanup
----------------------------
revision 1.14
date: 2000/07/18 08:42:08;  author: yur;  state: Exp;  lines: +4 -4
ve.c compile fix
----------------------------
revision 1.13
date: 2000/07/18 08:02:54;  author: yur;  state: Exp;  lines: +6 -1
namei.c: follow_dotdot() chroot vulnerability fix
----------------------------
revision 1.12
date: 2000/07/17 14:39:42;  author: den;  state: Exp;  lines: +5 -0
Root inode storage
----------------------------
revision 1.11
date: 2000/07/13 14:31:21;  author: den;  state: Exp;  lines: +104 -8
VeLink and configuration syscall support
----------------------------
revision 1.10
date: 2000/07/07 18:07:33;  author: den;  state: Exp;  lines: +48 -60
Major cleanup & sysctl demux patch
----------------------------
revision 1.9
date: 2000/07/06 11:29:35;  author: den;  state: Exp;  lines: +136 -155
VE link interface is implemented & procfs update
----------------------------
revision 1.8
date: 2000/06/29 12:37:11;  author: den;  state: Exp;  lines: +15 -49
Reboot allow patch
----------------------------
revision 1.7
date: 2000/06/27 19:45:04;  author: den;  state: Exp;  lines: +1 -3
Hostname patch update
----------------------------
revision 1.6
date: 2000/06/27 18:53:33;  author: den;  state: Exp;  lines: +19 -2
Hostname multiplication
----------------------------
revision 1.5
date: 2000/06/26 09:06:27;  author: den;  state: Exp;  lines: +86 -138
New edition of VE quit
----------------------------
```

```
revision 1.4
date: 2000/06/20 16:43:43;  author: den;  state: Exp;  lines: +7 -4
branches:  1.4.2;

Fixed capability issues, i.e. network interface can't be downed inside VE
----------------------------
revision 1.3
date: 2000/06/20 15:45:19;  author: den;  state: Exp;  lines: +0 -22

envid based routing
----------------------------
revision 1.2
date: 2000/06/19 14:48:12;  author: den;  state: Exp;  lines: +17 -4

ReiserFS + networking device is removed. Should work via alias
----------------------------
revision 1.1
date: 2000/06/18 09:04:04;  author: den;  state: Exp;
branches:  1.1.1;

Got ve patch from other repository
----------------------------
revision 1.1.1.2
date: 2000/08/14 16:24:57;  author: yur;  state: dead;  lines: +0 -0
Removed errneous files from vendor branch
----------------------------
revision 1.1.1.1
date: 2000/08/14 14:00:27;  author: yur;  state: Exp;  lines: +480 -439
2.4.0-test6_reiserfs-3.6.12
----------------------------
revision 1.4.2.2
date: 2000/06/21 13:59:59;  author: den;  state: dead;  lines: +0 -0
a
----------------------------
revision 1.4.2.1
date: 2000/06/21 13:07:46;  author: den;  state: Exp;  lines: +0 -890
ve.c added
----------------------------
revision 1.43.2.4
date: 2001/02/19 15:07:39;  author: den;  state: Exp;  lines: +4 -4
Finished 2-level routing. Before merge
----------------------------
revision 1.43.2.3
date: 2001/02/10 17:26:45;  author: den;  state: Exp;  lines: +20 -0
Compile time bugfixes
VE down support
----------------------------
revision 1.43.2.2
date: 2001/02/09 19:00:45;  author: den;  state: Exp;  lines: +34 -10
New 2-level routing scheme support
----------------------------
revision 1.43.2.1
date: 2001/02/02 22:08:36;  author: den;  state: Exp;  lines: +35 -8
veip hash support
=============================================================================
```

Exhibit C

From: "Thiele, Alan R." <AThiele@jenkens.com>
To: <tor@asplinux.ru>
Date: 1/8/01 7:46PM
Subject: COMMENTS

ALEXANDER...

Finally got a chance to look at your materials today so that I can provide more specific answers to your questions. Thank you for your patience.

RE: Method of effective reusing of common template file system tree in an environment with concurrent access and separate private modification area

Your question...Whether it is possible to apply it as full patent (non-provisional)?

My Answer...No. Here's why...Your Detailed Description of the Invention would not meet the standards of 35 U.S.C. 112, first paragraph as applied by the Examiners at the U.S. Patent and Trademark Office. Also you claims do not follow the standard form of preamble, transition and elements. But since 35 U.S.C. 112, first paragraph and the standard form of claims do not apply to provisional patent applications, we could your file work as a U.S. Provisional Patent Application with a little polishing up. Such polishing up would include putting reference numbers on your drawing figure and referring to those reference numbers in the specification.

RE: Distributed highly scalable wide area peer to peer network data storage with uniform name space and self-optimized data delivery model.

Your comment...the description of the invention will be provided in a few days

My response...if we file this as a provisional patent application, the key to an effective provisional patent application is to include as much as possible about the invention

RE: Utilization of cluster of computers with automatic configuration and virtual environments integrated with distributed file system as application service providing platforms

Your comment...the description of the invention will be provided in a few days

My response...if we file this as a provisional patent applicaiton, the key to an effective provisional patent application is to include as much as possible about the invention

RE: Virtual environment as a way of providing full independent operation system services with effective resource sharing and common single operation system kernel running on a single hardware node

Your question...What kind of tuning is required? what information do I have to supply?

My answer...As indicated above the specification may not be sufficient to meet the disclosure requirements of 35 U.S.C. 112, first paragraph. Specifically, reference numbers will have to be added and then included in the description of the embodiments. Additionally, the claims will have to be re-written to comply with U.S. standards. One thing you may not be aware of is that U.S. patents are unlike patents elsewhere in the world in that courts do not look to the core or the heart of the invention and then expand it later. In this country a good solid patent will describe then invention in great detail and the claims at the end will reflect that detail. This writing technique has recently been underlined by the Court of Appeals for the Federal Circuit who just recently severly limited the application of the court-created doctrine of equivalents for interpreting claims. This places a heavy burden on inventors and their attorneys to thoroughly describe inventions in as many embodiments and applications as possible. In effect, the Court of Appeals for the Federal Circuit is getting out of the business of trying to discern what was invented. Instead, the Court of Appeals for the Federal Circuit is effectively saying that if something isn't clearly spelled out in a patent, it isn't there. Bottom line is that patent applications have just become much harder to properly write effectively.

RE: Method of storing and retrieving of information with controllable redundancy for fault tolerant distributed storage

Your response...figs 1,2,3,and 4 were sent.

My comment...see my comments on the Virtual environment as a way of... patent application above. By the way I haven't received the figures from Jack yet. As previously indicated, hard copies of the figures provide more detail and are better to use than figures transmitted over the internet.

Think about who will eventually read any patent that issues. First, and actually least, is the patent examiner. In fact, past experience teaches that examiners only really pay attention to the claims and just skim the specification. Second, and of greater importance, is a potential buyer for your company. The patent should be a sales tool to convince a potential buyer that what you guys have invented is truly something special. A well written patent can add millions to the value of a business. Third, and of equal importance is the attorney for the potential buyer who will examine any issued patent in great detail to poke holes in its coverage, its quality and anything else to reduce the value of your business. Fourth, are the attorneys for potential infringers who will find any way to limit the scope and meaning of the claims at the end of your patent. Fifth, if you ever go to court, is the little old lady with blue tinted gray hair and tennis shoes who may sit on the jury to determine what your patent really means...you have got to write for her too. And these five are just for starters...consider potential licensees and their attorneys. All of this can be summed up in a comment made by U.S. Supreme Court many years ago to the effect that a properly written patent application is the most difficult document to write properly in all of American jurisprudence. And as indicated above, the Court Of Appeals For The Federal Circuit just made it a lot harder.

Hope this helps. Looking forward to hearing back from you.

art

- JENKENS & GILCHRIST E-MAIL CONFIDENTIALITY NOTICE -
This transmission may be: (1) subject to the Attorney-Client Privilege, (2) an attorney work product, or (3) strictly confidential. If you are not the intended recipient of this message, you may not disclose, print, copy or disseminate this information. If you have received this in error, please reply and notify the sender (only) and delete the message. Unauthorized interception of this e-mail is a violation of federal criminal law.

Exhibit D

From: "Alexander Tormasov" <tor@asplinux.ru>
To: "Thiele, Alan R." <AThiele@jenkens.com>
Date: 1/23/01 1:48PM
Subject: Re: COMMENTS

Hi, Alan!
Sorry for a delay.
some comment below. I want to consider first 2 most "ready" from my point of
view patents and want to bring them to
state suitable for Provisional patenting.
(for Azita - this is a comment about state of our application for 2 patent
for our new product called ASPcomplete).

Sincerely,
Alex Tormasov
ASPLinux Ltd,
ASPcomplete project
tor@asplinux.ru

> RE: Virtual environment as a way of providing full independent
> operation system services with effective resource sharing and common
single
> operation system kernel running on a single hardware node
>
> Your question...What kind of tuning is required? what information
> do I have to supply?
>
> My answer...As indicated above the specification may not be
> sufficient to meet the disclosure requirements of 35 U.S.C. 112, first
> paragraph. Specifically, reference numbers will have to be added and then
> included in the description of the embodiments. Additionally, the claims

what kind of reference numbers? for another patents or...?

> will have to be re-written to comply with U.S. standards. One thing you
may
> not be aware of is that U.S. patents are unlike patents elsewhere in the
> world in that courts do not look to the core or the heart of the invention
> and then expand it later. In this country a good solid patent will
describe
> then invention in great detail and the claims at the end will reflect that
> detail. This writing technique has recently been underlined by the Court
of

so, the climes should be much bigger in text?

> Appeals for the Federal Circuit who just recently severly limited the
> application of the court-created doctrine of equivalents for interpreting
> claims. This places a heavy burden on inventors and their attorneys to
> thoroughly describe inventions in as many embodiments and applications as
> possible. In effect, the Court of Appeals for the Federal Circuit is
> getting out of the business of trying to discern what was invented.
> Instead, the Court of Appeals for the Federal Circuit is effectively
saying

> that if something isn't clearly spelled out in a patent, it isn't there.
> Bottom line is that patent applications have just become much harder to
> properly write effectively.

bigger, more detailes or... I am not understand frankly speaking... what I
have to add here?

>
> RE: Method of storing and retrieving of information with
> controllable redundancy for fault tolerant distributed storage
>
> Your response...figs 1,2,3,and 4 were sent.
>
> My comment...see my comments on the Virtual environment as a way
> of... patent application above. By the way I haven't received the figures
> from Jack yet. As previously indicated, hard copies of the figures
provide
> more detail and are better to use than figures transmitted over the
> internet.
>
>
> Think about who will eventually read any patent that issues. First,
> and actually least, is the patent examiner. In fact, past experience
> teaches that examiners only really pay attention to the claims and just
skim
> the specification. Second, and of greater importance, is a potential
buyer
> for your company. The patent should be a sales tool to convince a
potential
> buyer that what you guys have invented is truly something special. A well
> written patent can add millions to the value of a business. Third, and of
> equal importance is the attorney for the potential buyer who will examine
> any issued patent in great detail to poke holes in its coverage, its
quality
> and anything else to reduce the value of your business. Fourth, are the
> attorneys for potential infringers who will find any way to limit the
scope
> and meaning of the claims at the end of your patent. Fifth, if you ever
go
> to court, is the little old lady with blue tinted gray hair and tennis
shoes
> who may sit on the jury to determine what your patent really means...you
> have got to write for her too. And these five are just for
> starters...consider potential licensees and their attorneys. All of this
> can be summed up in a comment made by U.S. Supreme Court many years ago to
> the effect that a properly written patent application is the most
difficult
> document to write properly in all of American jurisprudence. And as
> indicated above, the Court Of Appeals For The Federal Circuit just made it
a
> lot harder.
>

so, I am not clearly understand what we have to do for example with 2
mentioned above patents text.
We need to file them at least as provisional patents, so... what

polishing/rewriting/etc is required?

> Hope this helps. Looking forward to hearing back from you.
>
> art
>
>
> - JENKENS & GILCHRIST E-MAIL CONFIDENTIALITY NOTICE -
> This transmission may be: (1) subject to the Attorney-Client Privilege,
(2)
> an attorney work product, or (3) strictly confidential. If you are not the
> intended recipient of this message, you may not disclose, print, copy or
> disseminate this information. If you have received this in error, please
> reply and notify the sender (only) and delete the message. Unauthorized
> interception of this e-mail is a violation of federal criminal law.
>
>
>

CC: <azita@sw-soft.com>, <sb@sw.com.sg>

Exhibit E

From: "Thiele, Alan R." <AThiele@jenkens.com>
To: <tor@asplinux.ru>
Date: 1/27/01 6:34PM
Subject: PATENT APPLICATIONS

ALEX...

Sorry it took a few days to get back to you but it has been a real crazy week.

Here's where we are...

Re: Method of effective reusing of commom template file system tree in an environment with concurrent access and separate private modification area

COMMENT: As previously indicated, this disclosure could be filed as a U.S. Provisional Patent Application as I have the single figure and what you sent me includes claims, a desription of the prior art, a brief summary of the invention, a brief description of the figure and a detailed description of the invention.

The reason that this case is not ready to file as a Regular U.S. National application is that the claims would have to be re-written and reference numbers would have to be added to correlate parts of the drawing figure to what appears in the written portion of the application. For example, the written description of the invention calls for "carrier lines" and "subscriber lines". These "carrier lines" and "subscriber lines" appear in the drawing but it is hard to say which is which. To solve this problem the Rules of U.S. Patent Practice require that when an item is called out in both the written description of the invention and in the drawing figures it be given a reference number. Specifically, if the "carrier lines" were given the reference number 12, the number 12 would appear in the written description and on the drawing. That way the reader can point to exactly the item on the drawing that is being desribed in the written description of the invention.

The strict requirements in the Rules of Patent Practice for Regular U.S. National Patent Applications do not apply to U.S. Provisional Patent Applications and reference numbers are not required. However, I might add parenthetically that reference numbers would make the disclosure of the invention a lot easier to understand.

Based on the instructions in your e-mail of January 23, 2001, we will NOT file this as U.S. Provisional Application unless you give us specific authorization to do so.

Re: Distiributed highly scalable wide are peer-to-peer network data storage with uniform name space and self-optimized data delivery model

COMMENT: As previously indicated, all I have is a set of 6 claims. While the Rules of U.S. Patent Practice permit the filing of just about anything as a U.S. Provisional Patent application, I suggest that we flesh out the description of this invention before calling it a U.S. Provisional Patent application.

Re: Utilization of cluster of computers with automatic configuration and virtual environments integrated with distributed file system as application service providing platforms.

COMMENT: As previously indicated all I have is a set of 5 claims. While the Rules of U.S. Patent Practice permit the filing of just about anything as a U.S. Provisional Patent application, I suggest that we flesh out the description of this invention before calling it a U.S. Provisional Patent application.

Re: Virtual environments as a way of providing full independent operation system services with effective resource sharing and common single operation system kernel running on a single hardware node.

Let me first address the questions you asked in your e-mail message of January 23, 2001.

YOU ASKED..."what kind of reference numbers? for another patents or...?"

MY ANSWER...See my discussion of reference numbers above.

YOU ASKED..."so, the climes (claims?) should be much bigger in text?"

MY ANSWER...Probably. In U.S. Patents the claims are the most important part. It has always been this way; but, as I explained in my last memo the U.S. Court of Appeals for the Federal Circuit has just signficantly increased the importance of well written claims in a well thought out claim set. No longer will courts read a claim and determine if an infringer is effectively stealing the essence of an invention. According to the present state of the law it is now incumbent on the inventor and his attorney to fashion a set of claims that describes the invention in as many ways as possible. That set of claims should include claims which use the fewest words possible to explain the invention to detailed claims which clearly set out every aspect of the invention.

YOU ASKED..."bigger, more detailes or...I am not understand frankly speaking...what I have to add here?"

MY ANSWER...This response refers back to my previous answer. As I hope you now understand, more is better when building a set of claims. But in the U.S. we have a requirement called "antecedent basis". That is, everything that appears in a claim must have an antecedent basis in the description of the invention. To further understand the basis for this the U.S. Court of Appeals for the Federal Circuit seems to be getting very frustrated in trying to determine what the wording means in the claims of a U.S. Patent. To solve this problem they are shifting the burden of increasing the quality of patents back to inventors and their attorneys. This means that patent applications are harder to write in that they need to be more extensive in describing all the logical extensions of an invention. And all of the these logical extensions have to work their way into the claims. In effect, inventors and their attorneys now need to spend more time on thinking about how an infringer would try to copy an invention but make small differences to avoid the patent. Once this thinking process is complete these potential changes should be included into the written description of the invention and written into the claims.

COMMENT...My job is to obtain for you the best U.S. Patent I can. If I were to advise to file what you have sent to me, you may get a patent, but your patent wouldn't be worth much. A worthless patent can actually be a liability as it provides an undeserved sense of security. The real value of your patent is determined when your patent is subject to a detailed analysis when someone wants to buy your company or you decide to assert your patent against an infringer. Believe me, in these two situations, the claims in your patent will get a lot closer scrutiny than they ever get at the U.S. Patent and Trademark Office.

Certainly your disclosure is ready to file as a U.S. Provisional Patent Application but if the patent that you intend to obtain is to provide any value to SWsoft, it needs to be written for those future people who will be giving your patent close study when considering whether or not to buy your company or trying to find a way to invalidate the patented claims in an infringement situation.

Based on the instructions in your last e-mail, we will get this one ready to be filed as a U.S. Provisional Patent Application.

Re: Method of Storing and Retrieving of information with controllable redundancy for fault tolerant distributed storage.

First, I will address the question posed in your e-mail of January 23, 2001.

YOU ASKED..."so, I am not clearly understand what we have to do for example with 2 mentioned above patents text. We need to file them at least as provisional patents, so...what polishing/rewriting/etc is required.?"

MY ANSWER...My response above should answer this question.

COMMENT: Based on the instructions in your last e-mail, we will get this one ready to be filed as a U.S. Provisional Patent Application.

Have a good weekend.

art

CC: "Wiese, Bill" <BWiese@jenkens.com>